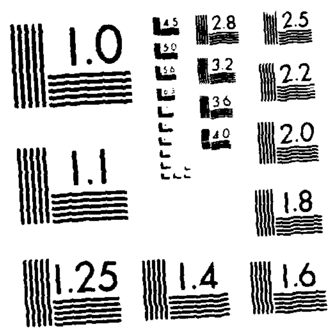
AD-A194 372 FINDING WHEELS OF VEHICLES IN STEREO IMAGES(U) ARMY ENGINEER TOPOGRAPHIC LABS FORT BELVOIR VA M W MCDONNELL ET AL. 21 AUG 87 ETL-R-141 1/1

UNCLASSIFIED F/G 14/4 NL

END
DATE
FILMED
8 88



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A194 372

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT: Approved for public release; distribution is unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S): R-141

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION: USAETL

6b. OFFICE SYMBOL (If applicable): CEETL-LO

6c. ADDRESS (City, State, and ZIP Code): Fort Belvoir, VA 22060-5546

7a. NAME OF MONITORING ORGANIZATION

7b. ADDRESS (City, State, and ZIP Code)

8a. NAME OF FUNDING / SPONSORING ORGANIZATION

8b. OFFICE SYMBOL (If applicable)

8c. ADDRESS (City, State, and ZIP Code)

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification): FINDING WHEELS OF VEHICLES IN STEREO IMAGES

12. PERSONAL AUTHOR(S): M.M. MCDONNELL, M. LEW

13a. TYPE OF REPORT: RESEARCH

13b. TIME COVERED: FROM ______ TO ______

14. DATE OF REPORT (Year, Month, Day): 87 AUGUST 21

15. PAGE COUNT

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number): model-based recognition, Hough transform, finding ellipses, stereo images, stereocorrelation

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

In model-based recognition of vehicles, wheels can play a key role. Certainly, they are the most prominent features of a vehicle. However, finding wheels in an image is a difficult task. Generally, a wheel appears as an ellipse in an image. An obvious way of finding ellipses is to use the Hough transform. The difficulty is that the search space is 5-dimensional. This curse of dimensionality is with us no matter what method we use to search for the ellipses. We use a stereo pair of images to reduce the search space. The idea is to determine from the stereo images the 3-D orientation of the plane containing the wheels, and then apply an appropriate transformation on either of the two stereo images such that in the new image the wheels will be circular. The search space is then only 3-dimensional. In this paper we describe this approach in detail and show some experimental results.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT: ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL: F. DARLENE SEYLER

22b. TELEPHONE (Include Area Code): (202) 355-2647

22c. OFFICE SYMBOL: CEETL LO

DD Form 1473, JUN 86 — Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE: UNCLASSIFIED

Finding wheels of vehicles in stereo images

M. M. McDonnell and M. Lew

U.S. Army Engineer Topographic Laboratories, Research Institute
Bldg. 2592, Fort Belvoir, Virginia 22060-5546

T. S. Huang

University of Illinois, Coordinated Science Laboratory
1101 W. Springfield Ave. Urbana, Illinois 61801-3082

1. ABSTRACT

In model-based recognition of vehicles, wheels can play a key role. Certainly, they are the most prominent features of a vehicle. However, finding wheels in an image is a difficult task. Generally, a wheel appears as an ellipse in an image. An obvious way of finding ellipses is to use the Hough transform. The difficulty is that the search space is 5-dimensional. This curse of dimensionality is with us no matter what method we use to search for the ellipses. We use a stereo pair of images to reduce the search space. The idea is to determine from the stereo images the 3-D orientation of the plane containing the wheels, and then apply an appropriate transformation on either of the two stereo images such that in the new image the wheels will be circular. The search space is then only 3-dimensional. In this paper we describe this approach in detail and show some experimental results.

2. THE PROBLEM AND AN APPROACH

In detecting and recognizing vehicles, wheels can play an important role. They are distinctive because of the rarity of circular forms in nature. Wheels usually appear as ellipses with the major axis approximately vertical, or at least perpendicular to the terrain slope. To find these ellipses in an image using Hough transform techniques we must search a 5-dimensional space.[1] The memory and time requirements of this search are so great that parallel implementations of the Hough transform described in the literature concentrate on finding straight lines, a much more tractable problem.

However, if we know the approximate orientation in 3-D of the plane containing the wheels, we can apply a geometrical transformation to the image such that the wheels appear as circles in the transformed image. Finding circles is much easier than finding ellipses since there are now only three degrees of freedom [the two coordinates of the center and the radius] rather than five and the search space is correspondingly only 3-dimensional rather than 5-dimensional. This is almost as easy, or as difficult, as finding straight line segments in an image.

Assume that a stereo pair of images is available. Then an obvious way to determine the wheel plane is stereocorrelation of a large number of points between the images, followed by fitting a plane to those points that represent wheels in the images. This has the disadvantage that you have to know where the wheels are in order to detect them! This is obviously not workable, so we propose and demonstrate a method which uses disparity values of the stereo pairs directly, without using triangulation, and finds best fits for planes. The assumption on which all of this rests is that groups of points falling near a plane are rare in nature and therefore indicative of artifacts such as vehicles.

We will present some relevant mathematics followed by a discussion of implementation considerations and finally the results of our experiments.

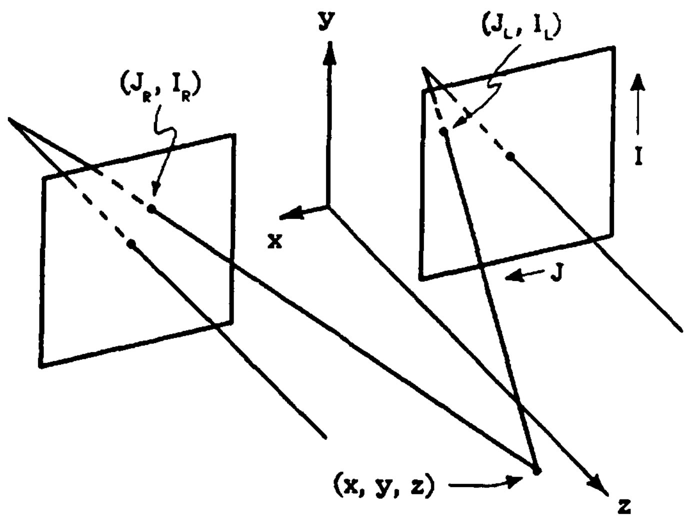


Figure 1. Stereo imaging geometry showing coordinate systems used.

3. ESTIMATING ORIENTATION OF A PLANE USING DISPARITY VALUES

Using the camera geometry and the notations indicated in Figure 1, we have the following triangulation formulas:

$$x = B\left(\frac{1}{2} + \frac{J_r}{J_l - J_r}\right) \qquad y = B\left(\frac{I_r}{J_l - J_r}\right) \qquad z = \frac{BF}{p}\left(\frac{1}{J_l - J_r}\right) + F \tag{1}$$

where

(x, y, z) = *3-D coordinates of a point in object space in meters.*

B = *Baseline of the camera system in meters.*

F = *Focal length of the cameras in meters.*

p = *Pixel spacing in meters.*

(J_r, I_r) = *2-D coordinates of a point in right image pixels measured from image center.*

(J_l, I_l) = *2-D coordinates of a point in left image pixels measured from image center.*

We now show that a plane in the x-y-z space corresponds to a plane in the J_r-I_r-Δ space, where

$$\Delta = J_l - J_r \tag{2}$$

is the disparity. Consider a plane in the x-y-z space:

$$ax + by + cz \doteq 1 \tag{3}$$

where a, b, c are real constants. Substituting (1) into (3), we get

$$\alpha J_r + \beta I_r + \gamma \Delta = 1 \tag{4}$$

where

$$\alpha = -\frac{p}{F}\frac{a}{c} \qquad \beta = -\frac{p}{F}\frac{b}{c} \qquad \gamma = \frac{p}{BF}\frac{1}{c} - \frac{p}{2F}\frac{a}{c} + \frac{p}{B} \tag{5}$$

Equation (4) clearly represents a plane in the J_r-I_r-Δ space.

Thus, if corresponding points in the stereo images are given, all of which lie near the wheel plane, then we can calculate disparity values Δ for these points and fit a plane over them in the J_r-I_r-Δ space. From this fitted plane, values of α, β, γ are obtained, and from (5) we get values for a, b, c.

4. VERTICAL PLANES

If a vehicle is moving on flat ground which is parallel to the x-z plane, then the wheel planes are most likely vertical, i.e. orthogonal to the x-z plane. In this case, $b = 0$ in (3), and (4) becomes

$$\alpha J_r + \gamma \Delta = 1 \tag{6}$$

which means that the plane in J_r-I_r-Δ space is orthogonal to the x-z plane. Therefore, the plane-fitting problem is reduced to a straight-line fitting problem in the J_r-Δ space.

Alternatively, we have from (6) and (2),

$$\alpha J_l + (\gamma - \alpha)\Delta = 1 \tag{7}$$

so we can fit a straight line in the J_l-Δ space.

5. EXPERIMENTAL RESULTS

The technique of Section 3 was applied to several stereo photographs. These were images of scenes containing a vehicle in an outdoor setting. We now show results from one such pair.

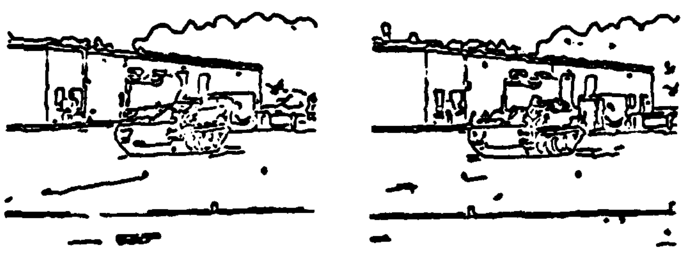


Figure 2. Edges from Canny operator

Figure 3. Extracted regions.

Figure 2 shows the edge map obtained from a stereo pair of images using Canny's operator.[2] A correlation algorithm based on the original grayscale images was used to find correspondences between the edge points of the left and right images. Disparity values were calculated, and points in the largest cluster of the disparity values were assumed to belong to the vehicle. These points are shown in Figure 3. Notice that most of these points lie either on the front face or on the side face (containing the wheels) of the vehicle.

Next the points of Figure 3 were mapped into the J_l-Δ space, resulting in Figure 4.

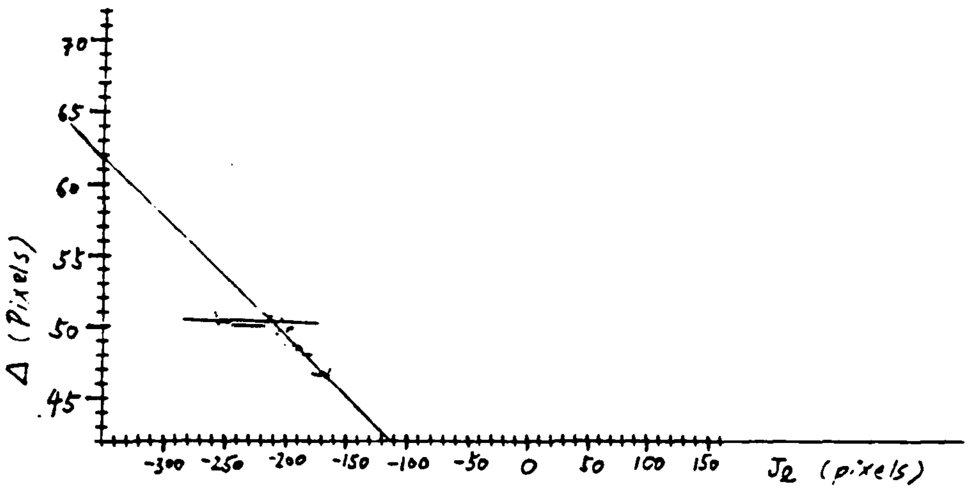


Figure 4. Disparity plot showing fitted lines.

Two straight lines were fitted to the points (by eye). The almost horizontal line in Figure 4 corresponds to the front face of the vehicle; the more slanted one corresponds to the side face containing the wheels. From inspection of Figure 4, the straight line corresponding to the wheel plane has the following equation:

$$\frac{J_l}{515} + \frac{\Delta}{33.2} = 1 \tag{8}$$

whence

$$\alpha = \frac{1}{515} \qquad \gamma - \alpha = \frac{1}{33.2}$$

Using (5) we get

$$\alpha = -\frac{p}{F}\frac{a}{c} = \frac{1}{\cdot 515}$$

$$\gamma - \alpha = \frac{p}{BF}\frac{1}{c} + \frac{p}{2F}\frac{a}{c} + \frac{p}{B} = \frac{1}{33.2}$$

For our camera setup

$$B = 2 \; meters \qquad p = 10^{-4} \; meter \qquad F = 0.1 \; meter$$

It then follows that

$$a = -0.031 \qquad c = 0.016$$

The equation for the wheel plane in the x-y-z space is

$$-0.031x + 0.016z = 1$$

The angle that this plane makes with the x-axis is

$$\theta = Tan^{-1}\frac{0.031}{0.061} = 63^{\circ}.$$

6. FINDING WHEELS

Once the equation for the plane in the x-y-z space is found, we can determine a geometrical transformation, based on the camera geometry, which will transform the left (right) image to a new image in which any ellipses on the plane before transformation become circles.

If the vehicle is relatively far away from the camera system, an orthographic approximation can be made. The transformation is then an inverse orthographic projection. In the case of a vertical plane such as in the example of section 4, this is simply:

$$J_l \rightarrow J_l' = \frac{J_l}{\cos\theta} \qquad I_l \rightarrow I_l'$$

In the new image space, the J_l'-I_l' space, we look for circles using either Hough transform or template matching methods, which have been proven to be identical mathematically.[3]

7. EXPERIMENT

We decided to use a template-matching algorithm rather than a Hough transform because it is easier to write and debug, and because we already had many programs available from previous work that could be used in template matching. Following is an outline of the method used. Variables (A) through (F) are arrays and variable (L) is a list. All arrays have eight bits per element except for array (C) which is a 1-bit array.

1. Define a region of interest on one of the stereo photographs (A).
2. Stretch (A) by the amount indicated by plane-fitting to correlated points, as described above, to make stretched image (B).
3. Run Canny's edge-finder on (B) to make the edge image (C).
4. Make (C) into an 8-bit array (D).
5. Smooth (D) by Gaussian kernel convolution to get (E).
6. Run a circle mask over (E), starting with the smallest radius of interest, normalize the sum of masked values by the number of pixels in the circle [the mask area], and store the result at the location of the center of the circle in (F).
7. Find some number of peaks in (F) and store in a list (L).
8. Increment the circle radius and go to step 6. Repeat steps 6 through 8 until the range of circle radii of interest has been tried.
9. Search list (L) and display the best matches for chosen radii.

Canny's edge operator was used in step 3 and some of Canny's functions were used in step 5. The rest of the programming was done by us on Symbolics lisp machines.

Smoothing could be done on the match array (F) rather than on the edge array (C), but it is easier to visualize the result of smoothing the edge image so this was done. The major variable of interest in the procedure outlined above is the size of the kernel used for smoothing. Tests showed us that a kernel of radius 3 was the best for allowing circles to be picked out of noise. This kernel size was used for smoothing array (D) and for smoothing the image as part of the edge-finding procedure. Larger convolution kernels tend to mix things up too much so that many spurious circles are found, and smaller kernels allow noise to cause much the same thing, so there seems to be an optimum, though we suspect that it is difficult to decide *a priori* what that should be as the optimum width is probably scene-dependent. For convolution kernels of radius 3 we found it to be reasonable to sample circle radii in steps of 2 pixels at a time so a sample sequence of circle radii might be 6, 8, 10, ...

Several vehicle scenes from a series were operated on using the method described above. Some of our best results are shown in Figure 5 where the left image corresponds to array (A) in the algorithm description and the right image corresponds to array (B).

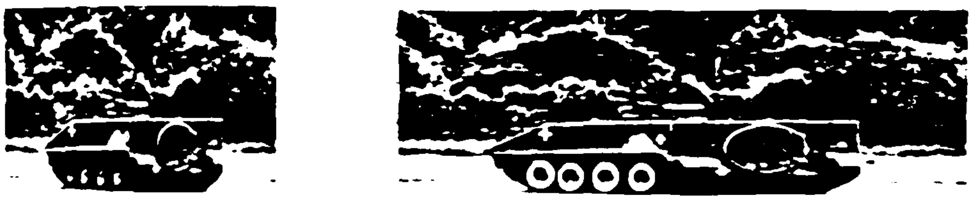


Figure 5. Original image on left. Stretched image on right with detected wheels overlayed. Ten hypothesized wheels are shown.

An integer algorithm[4] was used to perform the circle masking and a run on a 256 X 256 input image takes about 15 minutes for testing five different circle radii over the whole stretched image, which had a width of 435 for the example shown in Figure 5. As a matter of interest, a match array is shown in Figure 6. The peaks at the vehicle wheels are visible in Figure 6, but they are much more evident for an image displayed on a CRT.

Figure 6. Match array, a convolution of a wheel template and an edge plot.

We found that programming a template-matching algorithm was considerably easier than programming a Hough transform for the same task, though the Hough transform may have advantages in speed. Advantages of the Hough transform were not investigated, though some fruitless effort was spent attempting to program a Hough transform circle-matcher before we did the template-matcher. Speedups were also gained by working on only one circle radius at a time so that only a two-dimensional match array was in computer memory at a time rather than a three-dimensional one. Keeping a record of maxima of a given match array in list (L) was adequate for our purposes and reduced paging to a minimum by reusing the same 2-D match array for each radius.

Matching was not good for one of our images where the wheels were small, about 6 pixels in radius, and where the angle was large, around 60 degrees. This is the image for which the edge extraction and analysis is shown in figures 2, 3, and 4. Unfortunately, the limited time of our collaboration did not allow us to perform correlation on more suitable images, and the results shown in figures 5 and 6 are from estimation of the correct amount of stretching by eye rather than by correlation. The results seem interesting enough to present despite the lack of correlation on a good image. We have at least implemented all the techniques necessary for design of an automated wheel-finding system.

Although we did not have a test case to check this assertion, we believe that it was the large angle, necessitating horizontal stretching by more than a factor of two, that caused poor performance in this case, and not the small radius of the wheels or the image quality, which was also poorer for this extreme case [figures 2, 3,and 4] than for the other images we dealt with. As a rule, we think that stretching an image by more than about 1.6 will lead to progressively poorer performance. This is not too stringent a requirement since a stretch of 1.6 corresponds to a deviation angle of about 50 degrees away from a side view, allowing a wide range of angles to be accommodated. This is even less of a restriction when you realize that for very large angles, say above 80 degrees, the wheels will not be visible at all, and for smaller but still large angles the wheels may only be a couple of pixels wide in projection and the sides of the wheels will become significant in the view. All of these effects will confound the matching program, and would cause an equal amount of trouble if ellipse matching were done directly.

8. SUMMARY

A method using digital stereocorrelation and array resampling to reduce dimensionality in the automated search for wheels in a scene has been analyzed and demonstrated.

This work could be enhanced for further automation by searching for linear alignment of hypothesized wheel centers. Also, it is a difficult problem to fit several planes to scattered 3-D points, a problem which was finessed in this investigation by fitting the planes by eye. We have used linear anamorphic magnification rather than a full perspective transformation, though we feel that this approximation is valid for realistic cases. Despite these difficulties, we think that this work is important as it stands since it demonstrates a workable and refinable method for performing an important task in automated scene analysis.

Source code for the array resampling and circle template matching is available for ftp on the Arpanet to interested parties. It is written in Common Lisp to run on Symbolics lisp machines. Contact McDonnell [Arpanet: mike@etl.arpa] for details.

9. REFERENCES

[1]. D. H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes," Pattern Rec. 13(2), 111-122 (1981).

[2]. J. F. Canny, "A Computational Approach to Edge Detection," IEEE PAMI, PAMI-8(6), 679-698,

(1986).

[3]. G. C. Stockman and A. K. Agrawala, "Equivalence of Hough Curve Detection to Template Matching," Commun ACM, 20(11), 820-822, (1977).

[4]. J. Bresenham, "A Linear Algorithm for Incremental Digital Display of Circular Arcs," CACM, 20, 100-106 (1977).

ATE

LMED

8